Лабораторная работа № 3

студента группы ИТз-221

Дмитриева Дмитрия Анатольевича

*Выполнение: \_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_*

Основы объектно-ориентированного программирования

*Цель работы***:** знакомство с основными принципами объектно-ориентированного программирования и средствами их реализации в C#. Приобретение практических навыков при разработке консольного приложения для работы с классами.

**Ход работы:**

***Вариант - 12***

1. По заданному описанию класса в соответствии с вариантом создал класс на языке C#. Класс Time для работы со временем в формате «час:минута:секунда»
2. Поля: unsigned int h, m, s // часы, минуты и секунды (рис. 1)
3. Методы и перегрузки операций:
   * вывод на экран (рис. 2);
   * вычисление разницы между двумя моментами времени в секундах (рис. 3);
   * сложения и вычитание времени из заданного количества секунд (рис. 4);
   * сравнение моментов времени (рис. 5);
   * перевод в секунды и минуты (рис. 6);
   * перегрузка конструкторов (рис. 7);
   * перегрузка ToString (рис. 8).

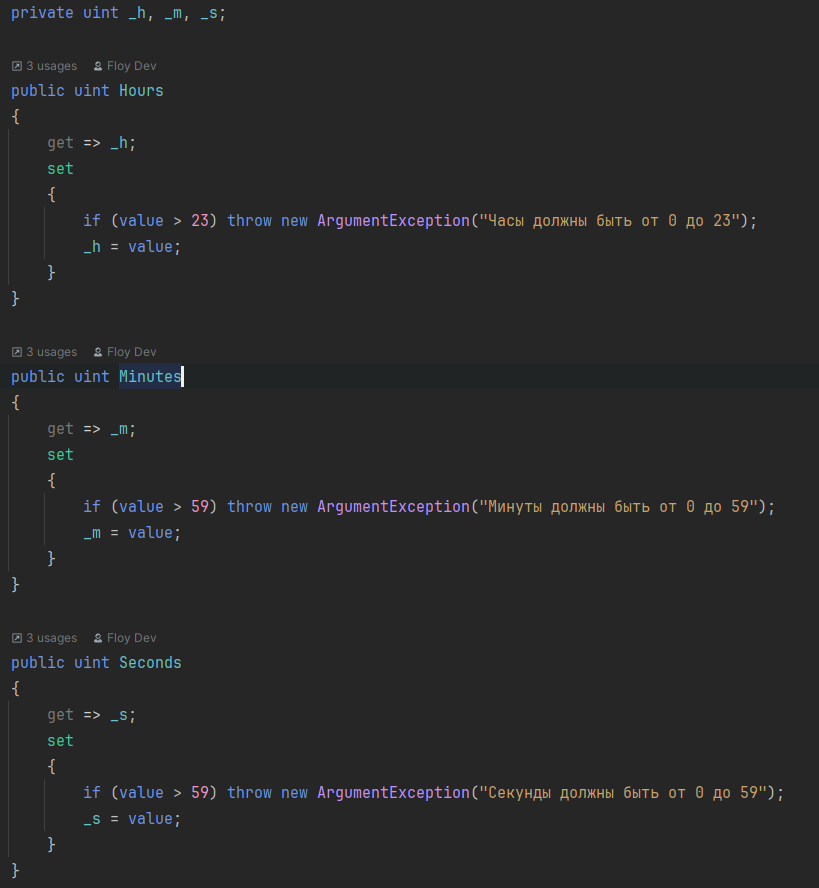


Рисунок 1 – беззнаковые поля

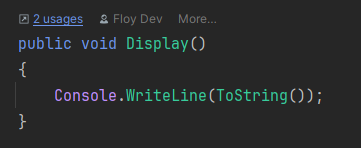
****

Рисунок 2 – вывод времени

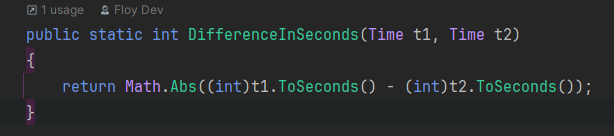


Рисунок 3 – разница между двумя моментами времени в секундах

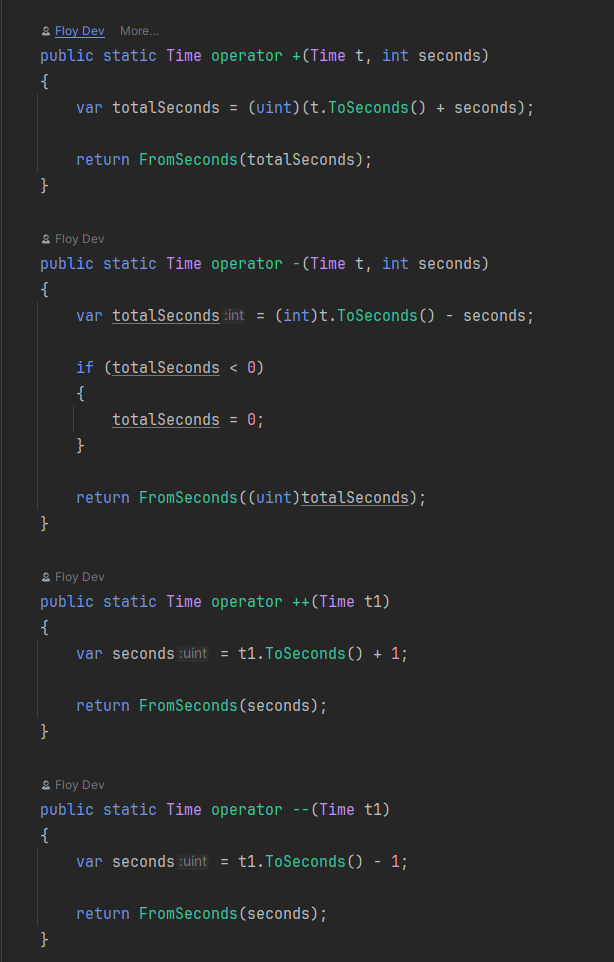


Рисунок 4 – сложения и вычитание времени

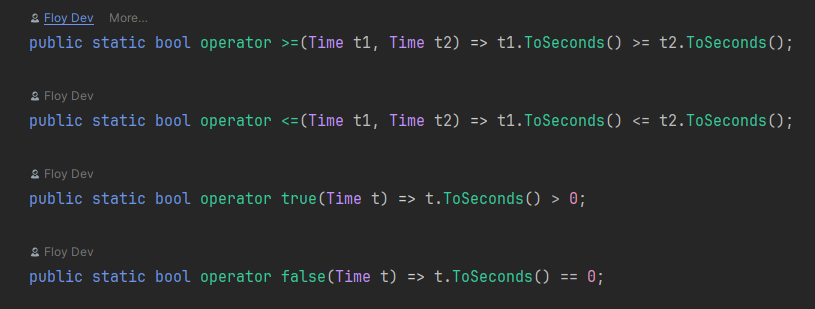


Рисунок 5 – сравнение моментов времени

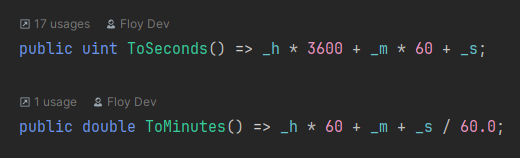


Рисунок 6 – перевод в секунды и минуты

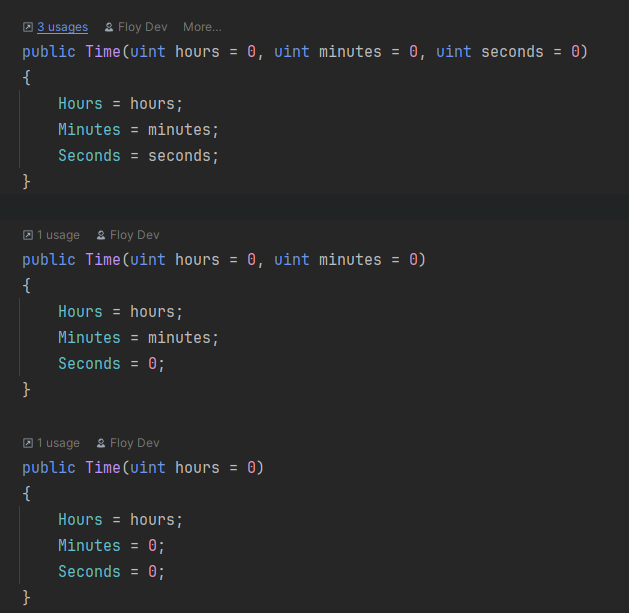


Рисунок 7 – перегрузка конструкторов

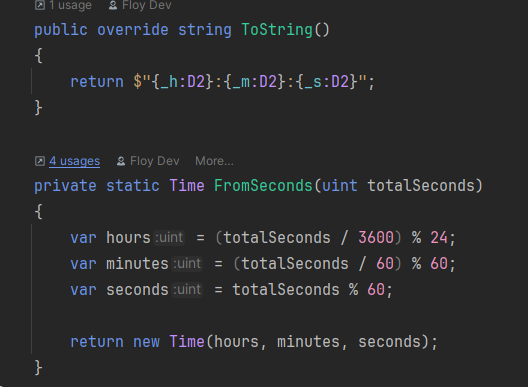
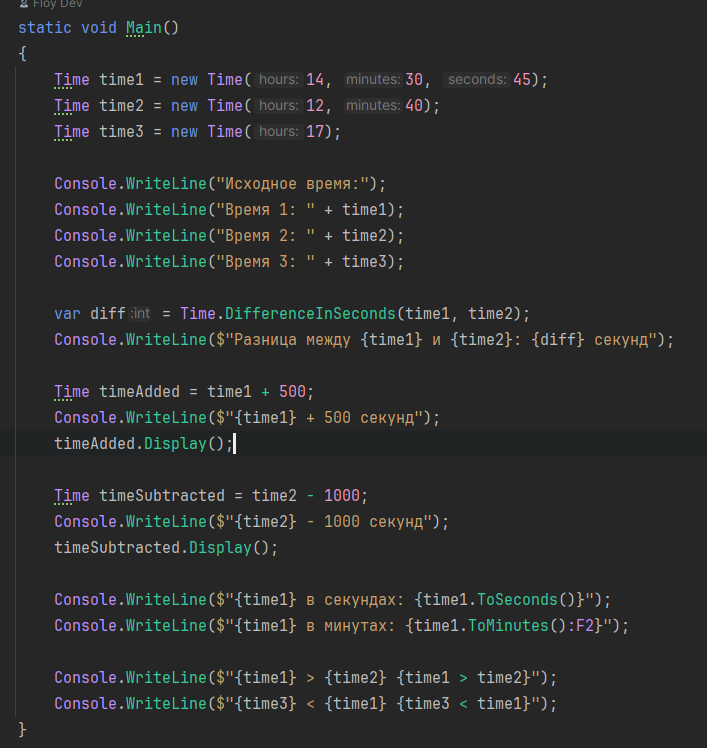


Рисунок 8 – перегрузка метода ToString

1. Протестировал программу, получил следующий результат (рис. 9)



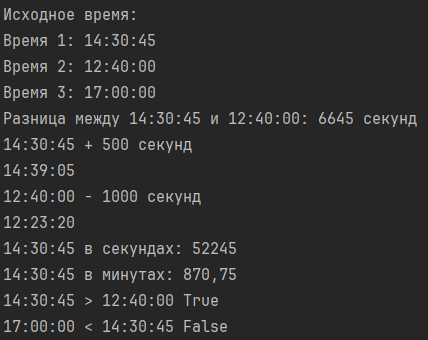
****

Рисунок 8 – результаты работы программы

**Контрольные вопросы**:

1. Понятие ООП и основные принципы: ООП (Объектно-Ориентированное Программирование) — это парадигма, в которой объекты взаимодействуют друг с другом, а данные инкапсулируются. Основные принципы: инкапсуляция, наследование, полиморфизм.
2. Понятие класса и объекта: Класс — это шаблон для создания объектов, описывающий их состояние и поведение. Объект — это экземпляр класса.
3. Члены класса, их назначение: Члены класса — это поля, методы, свойства и события. Они определяют поведение и состояние объекта.
4. Полиморфизм: Полиморфизм позволяет объектам разных типов реагировать на одинаковые сообщения (методы) по-разному.
5. Наследование: Наследование позволяет создавать новые классы на основе уже существующих, повторно используя их функциональность.
6. Инкапсуляция: Инкапсуляция скрывает внутренние детали реализации объекта и предоставляет доступ к его состоянию только через публичные методы.
7. Описание классов в C#: Классы в C# описываются с помощью ключевого слова class, могут включать поля, методы, свойства и конструкторы.
8. Модификаторы доступа классов: Классы могут иметь модификаторы доступа public, internal, protected internal, private, ограничивающие доступ к ним из других частей программы.
9. Модификаторы доступа членов класса: у членов класса могут быть модификаторы public, private, protected, internal, определяющие доступность для внешнего кода.
10. Перегрузка методов: Перегрузка методов позволяет создавать несколько методов с одинаковым именем, но с разными параметрами.
11. Перегрузка операций: Перегрузка операций позволяет определять собственное поведение стандартных операторов (например, +, -) для пользовательских типов.
12. Основные принципы ООП: это инкапсуляция (скрытие данных), наследование (создание новых классов на основе существующих) и полиморфизм (обработка объектов одинаковым образом, несмотря на их различия).
13. Статические классы и поля: Статические классы не могут быть инстанциированы, а их члены доступны через сам класс. Статические поля принадлежат самому классу, а не его объектам.
14. Авто заполняемое свойство: это свойство, у которого автоматически создается скрытое поле, управляемое системой, с возможностью только получения или установки значения.
15. Параметры методов: Параметры методов — это данные, которые передаются в метод для обработки. Они могут быть обязательными или опциональными (с дефолтными значениями).

**Вывод**: ознакомился с основными принципами объектно-ориентированного программирования и средствами их реализации в C#. Приобрел практические навыки при разработке консольного приложения для работы с классами.